

**Amendments to the Specification:**

Please amend the specification as follows:

**Page 4, lines 2-23 to Page 5, lines 1-5:**

ycc  
6/7/07

The embodiments of the present invention fill these needs by providing an Extensible Markup Language (XML) based report generator. The XML based report generator of the embodiments of the present invention allows a test summary report to be generated from a test execution log file quickly, generally without manual intervention from a user, and consequently, reducing human induced errors. In one embodiment, a method for creating a test summary report is disclosed. Broadly speaking, a test is executed and the test results are generated in [[a]] an XML enabled format. The XML enabled test results are processed to create a test summary report.

In another embodiment, [[a]] an XML based report generator is disclosed. The XML based report generator includes a parser that processes a test execution log file to generate a well-formed XML test reports file. In addition, a logical parser is included that processes the well-formed XML test reports file to produce a logically arranged XML test reports file. The XML based report generator further includes a Hypertext Markup Language (HTML) converter parser that converts the logically arranged XML test reports file into [[a]] an HTML test summary file.

Another method for creating a test summary report is disclosed in a further embodiment of the present invention. The method includes executing a test application on a platform, where the test application is executed using a status utility having functions that generates XML code. The test results are generated in [[a]] an XML enabled format using the status utility, and are output to a test execution log file. The test execution log file is processed to generate a well-formed XML test reports file, which is then arranged to create a

logically arranged XML test reports file. The logically arranged XML test reports file is then converted into [[a]] an HTML test summary report. Other aspects and advantages of the invention will become apparent from the following detailed description, taken in conjunction with the accompanying drawings, illustrating by way of example the principles of the invention.

**Page 7, lines 2 to 10:**

An invention is disclosed for [[a]] an XML based report generator. The XML based report generator of the embodiments of the present invention allows a test summary report to be generated from a test execution log file quickly, generally without manual intervention from a user, and consequently, reducing human induced errors. In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without some or all of these specific details. In other instances, well known process steps have not been described in detail in order not to unnecessarily obscure the present invention.

**Page 8, lines 1-17 to page 10, lines 1-5:**

ycc  
6/7/07

Each test listing 208a-208b lists compile test results 210a-210b, execute test results 212a-212b, and a test result 214a-214b. The compile test results 210a-210b list information on the test compilation for the particular test 208a. For example, the compile test results 210a can list whether or not the test 208a compiled correctly, and if not, source code errors. The execute test results 212a-212b list information on the test execution of the particular test 208a. For example, the execute test results 212a can list whether or not the test 208a executed correctly, and if not, the reason for the execution failure. The test results 214a-214b list the actual test output for the particular test. For example, test result 214a can list the actual test output for test 208a, including whether the test passed or failed, and in the case of

504. In addition, the parser 502 can extract control characters not utilized during further operations of the process.

**Page 13, lines 8 to 21:**

As mentioned above, the parser 502 creates a well-formed XML test report file 504. In addition, embodiments of the present invention create the well-formed XML test report file 504 such that the XML enabled test reports are valid as well, according to the Test DTD. A well-formed XML document is [[a]] an XML document that complies with XML well-formedness constraints. These constraints require that elements, which are named content containers, properly nest within each other and use other markup syntax correctly. Unlike HTML, well-formed XML elements are defined by their use, not by a rigid structural definition, allowing authors to create elements in response to their development. A valid XML document is [[a]] an XML document that conforms with a corresponding DTD. As mentioned above, [[A]] a DTD is a set of rules that a document follows, which software may need to read before processing and displaying a document. These rules generally state the name and contents of each element and in which contexts it can exist. Paragraph elements might be defined as containing keyword and code elements and as existing within section and note elements.

**Page 14, lines 4-23 to page 15, lines 1-5:**

7/6/07

As each test is executed, the results are written to the test execution log file 200 using the status utility, as described above with reference to Figure 4. However, during a particular test cycle, a test engineer may run any number of tests that are available in a particular test suite. That is, although a particular test suite may include a predefined number of tests, the test engineer may run all, some, or none of the tests within the particular test suite. Thus, the actual number of test executed in a particular test suite may not be known until the test suite is actually executed. The embodiments of the present invention address this issue by writing